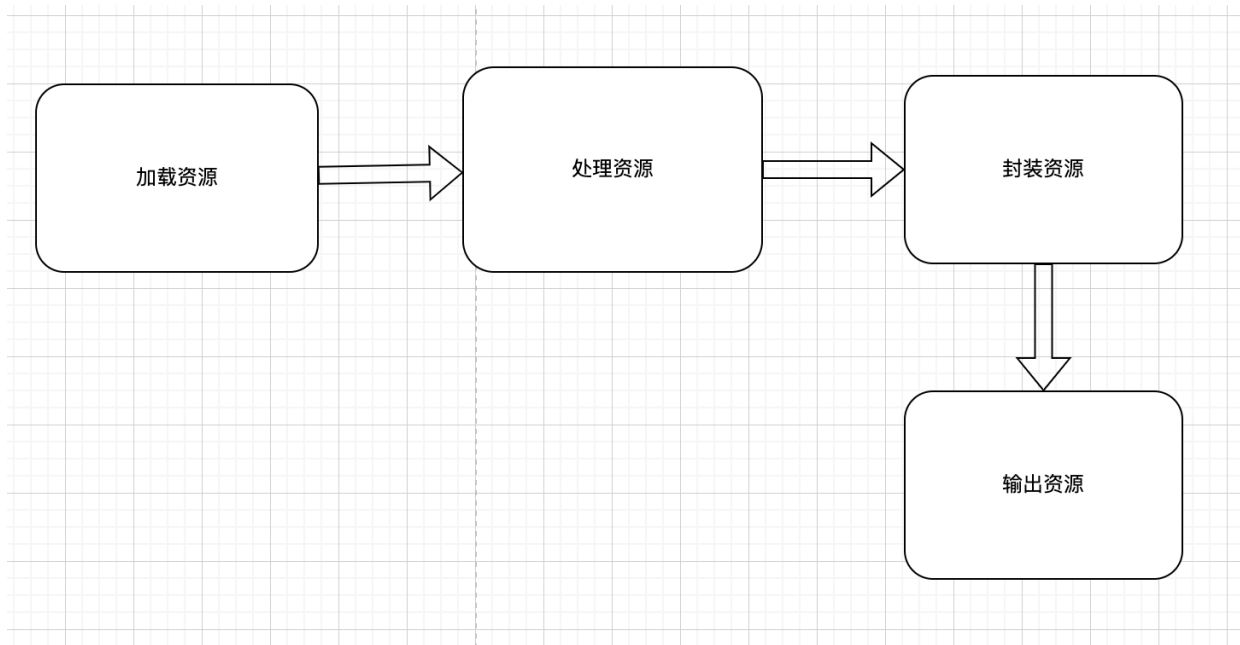


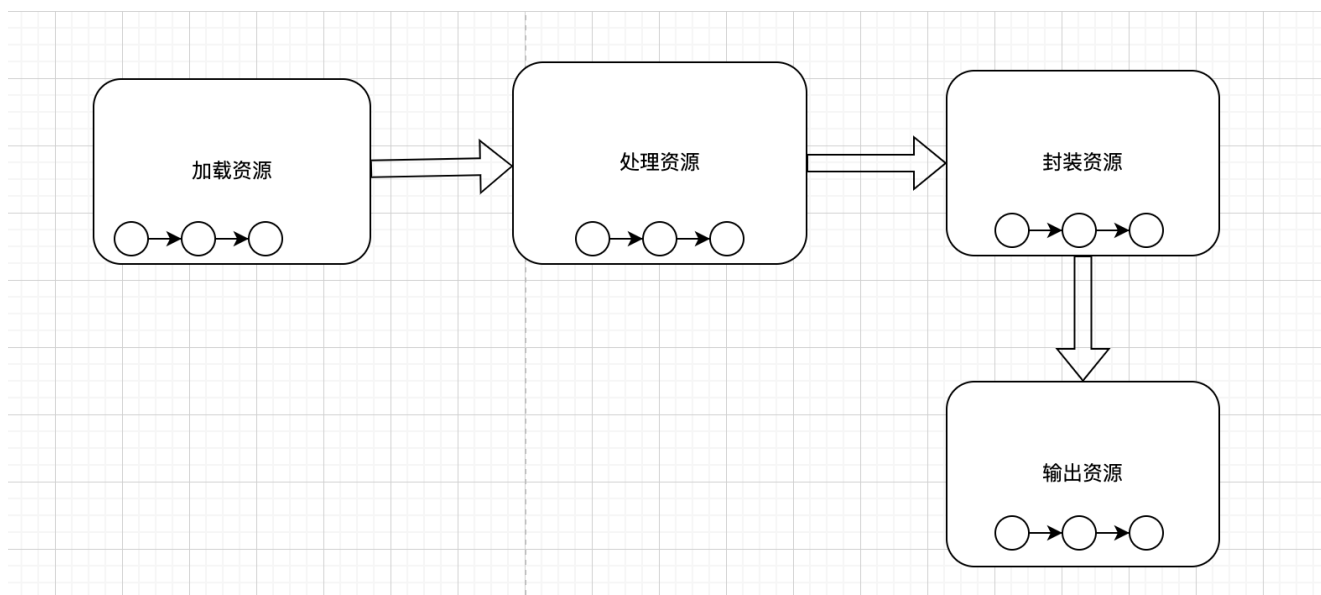
# webpack的简单 workflow

webpack的工作流程是固定的，可以简单的归为：

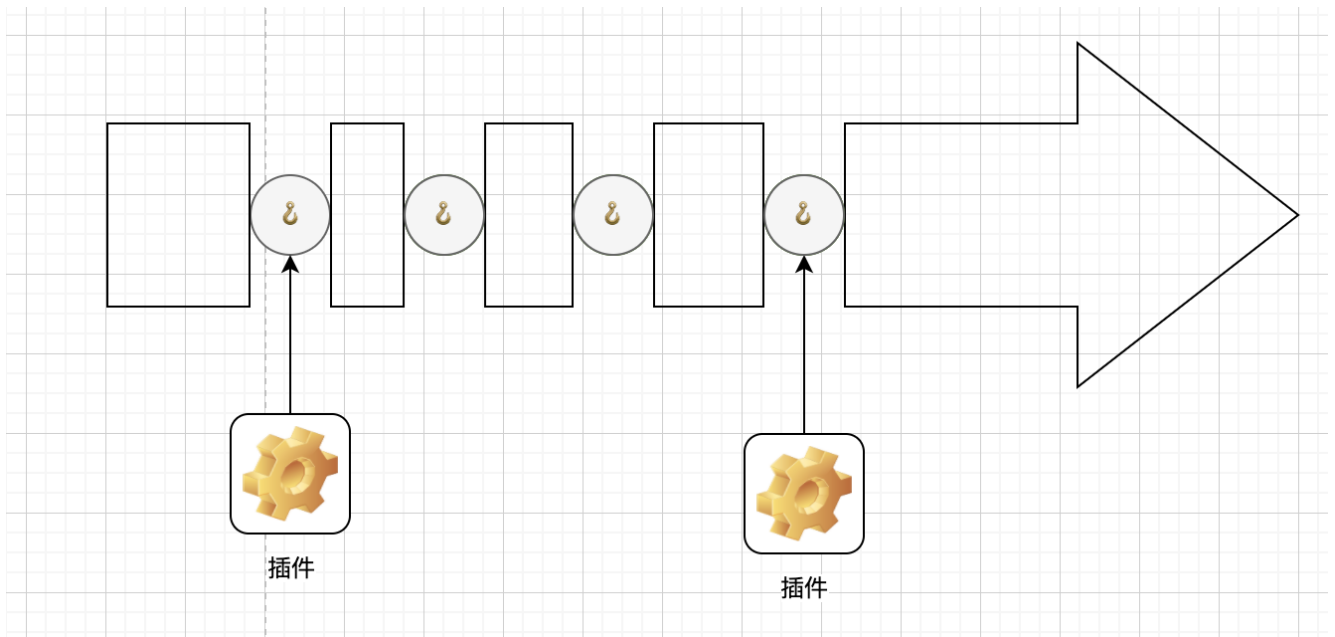
1. 加载资源
2. 处理资源
3. 封装资源
4. 输出资源



这是一个非常非常宏观的角度了解webpack，其中的每一个大的步骤都有十分复杂内部流畅，所以上述的图片可能会变成如下，每个小的圆圈都表示特定的工作任务。  
webpack为在每个一个“任务”附近加入了很多的钩子，可以勾入需要要特定节点要做的一些“额外的任务”。这些“额外”的任务就是plugins。



如果webpack的整个工作流融为一体，那么，钩子就存在于整个工作流的特定时间节点，插件的意义就是将自身与这个时间节点挂钩。



## 一些思考

1. plugin的概念很多且复杂，但是背后的理念思想却十分的简单，复杂的是API接口。
2. 学习plugin，应该把背后的设计理念先搞清楚，再去研究API，这时候学习的时候心里更有底气，知道学习的什么内容。
3. 简单分析后，不难发现，插件的机制践行的时候“消息的发布与订阅”，颇有“事件”机制的影子，其背后的设计理念也与AOP的思想有几分相似！
4. 设计模式真的很重要，不仅仅的开发中重要，掌握了设计模式，学习速度如同开挂～
5. 钩子简而言之就是，在合适的时候，做合适的事情。